

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

2
3
4
5

7

8
9
10
11
12
13
14
15
16

18

19
20
21

The Claims

Claim 1, as amended, recites a method for testing at least one software application, the method comprising:

- **installing at least one hook function into an application programming interface (API) of an operating system, the at least one hook function configured to monitor operating system messages communicated with the software application during execution of the software application being tested;**
- **retrieving information** descriptive of a state of operation of the software application being tested and at least one graphics element rendered during execution of the software application being tested, wherein the information identifies an executable feature associated with the at least one graphics element, and **wherein at least some of the retrieved information descriptive of the state of operation is based on messages monitored by way of the at least one hook function;**
- storing information related to an association between the executable feature and the at least one graphics element and the state of operation of the software application in a map data structure containing information related to at least one graphics element for testing, the association and information being stored in the map data structure during execution of the software application being tested;
- automatically selecting an executable feature from the map data structure based on the association stored in the map data structure;
- automatically executing the selected executable feature associated with the graphics element; and
- dynamically updating the information related to the state of operation of the software application and the association in the map data structure upon execution of the executable feature.

Applicant respectfully disagrees with and traverses the rejection of claim 1, as amended.

Specifically, Parker fails to teach or suggest installing at least one hook function into an application programming interface (API) of an operating system,

1 the at least one hook function configured to monitor operating system messages
2 communicated with the software application during execution of the software
3 application being tested, as recited by claim 1, as amended. Also, Parker fails to
4 teach or suggest retrieving information...., wherein at least some of the retrieved
5 information descriptive of the state of operation is based on messages monitored
6 by way of the at least one hook function, as recited by claim 1, as amended.

7 Rather, Parker is directed to testing new and revised computer
8 applications which use a graphical user interface (GUI), wherein a *test script*
9 written in a high-level programming language is used in conjunction with a test
10 executive and a test driver (Abstract of Parker). As to hook functions, Parker
11 makes only the following casual mention:

12
13 A "wrapper" is used to intercept calls to a target procedure so that
14 some additional action can be performed whenever the target
15 procedure is called. Before or after performing the additional
16 action, the wrapper actually calls the target procedure. In the X
17 environment, the test tool of the present invention uses wrappers
18 around selected routines to intercept calls which the Application,
19 300, makes to the LSEM, 305, over interface 301 and then routes
20 the information contained in the application's calls over interface
21 323 to the test tool. An alternative technique for monitoring the
22 Application, 300, to GUI, 307, interface is to make use of hook
23 mechanisms built into a GUI. Microsoft Windows.TM. provides
24 such hooks. Timers in the test tool can be used to poll the hooks at
25 regular intervals. (Col. 12, lines 36-49 of Parker).

21 Parker provides no information regarding how a hook is used or operates,
22 nor any information regarding what messages are or can be monitored by way of
23 a hook (or hooks). Also, Parker is completely lacking any material related to
24 installing a hook in any way, not the least of which being with the particularity
25 recited by claim 1, as amended. Parker is directed to the use of a "wrapper" and

1 is satisfied with that distinctly different technique. For at least the foregoing
2 reasons, Parker fails to teach or suggest all of the features and limitations as
3 positively recited by claim 1, as amended.

4 Santee fails to cure the deficiencies of Parker. Specifically, Santee fails to
5 teach or suggest installing at least one hook function into an application
6 programming interface (API) of an operating system, the at least one hook
7 function configured to monitor operating system messages communicated with
8 the software application during execution of the software application being
9 tested, as recited by claim 1, as amended. Also, Santee fails to teach or suggest
10 retrieving information..., wherein at least some of the retrieved information
11 descriptive of the state of operation is based on messages monitored by way of
12 the at least one hook function, as recited by claim 1, as amended.

13 Rather, Santee is directed to mapping a GUI of an application through
14 recursive manipulation of that GUI (Col. 2, lines 28-42 of Santee). Santee
15 provides that such mapping includes comparing a first set of windows to a
16 second set of windows so as to identify a new window within the second set
17 (*Id.*). However, Santee is totally devoid of the terms “hook”, “hooks”, or any
18 respective equivalents. In turn, Santee is lacking any mention of installing a
19 hook in any way or for any purpose.

20 There is no way to select elements from Parker, and then to somehow
21 combine those elements with other elements selected from Santee, in order to
22 arrive at the subject matter of claim 1, as amended, as no possible combination
23 of Parker and Santee teaches or suggests all of the required features and
24 limitations. In view of the foregoing reasons, as well as for other reasons argued
25 previously in prosecution, Applicant asserts that the § 103 rejection of claim 1,

1 as amended, is unsupportable and should be withdrawn. Applicant further
2 asserts that claim 1, as amended, is allowable.

3 **Claims 2-6, 8-11 and 40**, as respectively amended, are also allowable as
4 depending from an allowable base claim. While the respective § 103 rejections
5 against claims 2-6, 8-11 and 40 have been fully considered, they are not seen as
6 contributing anything of merit.

7 In further regard to claim 2 (as amended), that claim includes all of the
8 features and limitations of claim 1 (as amended) from which it depends, as well
9 it's own particularly recited subject matter. For example, claim 2 recites, among
10 other things, dynamically updating information in the map data structure
11 descriptive of at least one *second graphics element* resulting from the exposure
12 of a *new state of operation* of the software application in response to the
13 execution of the executable feature.

14 More to the point, claim 2, as amended, recites subject matter that builds
15 upon and continues the method subject matter of claim 1, as amended. Applicant
16 asserts that claim 2, as amended, is exemplary of at least some of the distinctions
17 between the subject matter of the pending application, and the respective
18 teachings of Parker and Santee.

19 **Claim 13**, as amended, recites a system for generating a map, comprising:

- 20
- 21 • a capture agent for retrieving information descriptive of a state of
22 operation of a software application being tested and a plurality of
23 graphics elements rendered during execution of the software
24 application, the information including an executable feature
25 associated with each graphics element, **the capture agent**
configured to install at least one hook function into an
application programming interface (API) of an operating
system, wherein the at least one hook function is configured to

1 **monitor messages communicated between the operating system**
2 **and the software application during execution of the software**
3 **application being tested;**

- 4 • an application driver for storing information in a map data structure
5 related to an association between each executable feature and
6 corresponding graphics element and a state of operation of the
7 software application during execution of the software application
8 being tested, wherein the map data structure contains information
9 related to at least one graphics element for testing;
- 10 • an indicator for tracking a dynamic updating of the information an
11 application driver for automatically selecting one of the executable
12 features stored in the map data structure based on the information
13 stored in the map data structure;
- 14 • a command agent for automatically executing the selected
15 executable feature; and
- 16 • an indicator for tracking a dynamic updating of the information
17 related to the association and the state of operation of the software
18 application in the map data structure upon the automatic execution
19 of the selected executable feature.

20 Applicant respectfully disagrees with and traverses the rejection of claim
21 13, as amended.

22 Specifically, neither Parker nor Santee – whether consider alone or in any
23 properly motivated combination - teaches or suggests a capture agent configured
24 to install at least one hook function into an application programming interface
25 (API) of an operating system, wherein the at least one hook function is
26 configured to monitor messages communicated between the operating system
27 and the software application during execution of the software application being
28 tested, as positively recited by claim 13, as amended. As argued above, Parker
29 provides no specific teachings as to the use or operation of a hook, or any subject
30 matter related to installing a hook. In turn, Santee is totally devoid of any
31 mention of a hook, in any context. Thus, no possible combination of Parker and

1 Santee teaches or suggests the particular subject matter recited by claim 13, as
2 amended.

3 In view of the foregoing reasons, as well as for reasons argued previously
4 in prosecution, Applicant asserts that the § 103 rejection of claim 13, as amended,
5 is unsupportable and should be withdrawn. Applicant further asserts that claim
6 13, as amended, is allowable.

7 **Claims 14-15 and 17-20**, as respectively amended, are also allowable as
8 depending from an allowable base claim. While the respective § 103 rejections
9 against claims 14-15 and 17-20 have been fully considered, they are not seen as
10 contributing anything of merit.

11 **Claim 22**, as amended, recites a method for systematically invoking an
12 executable feature of a software application having a graphical user interface, the
13 method comprising:

- 14 • **installing at least one hook function into an application**
15 **programming interface (API) of an operating system, the at**
16 **least one hook function configured to monitor operating system**
17 **messages communicated with the software application during**
18 **execution of the software application being tested;**
- 19 • retrieving information descriptive of a state of operation of a
20 software application being tested and at least one graphics element
21 rendered during execution of the software application, the
22 information including an executable feature associated with the at
23 least one graphics element, **at least some of the retrieved**
24 **information descriptive of a state of operation of the software**
25 **application retrieved by way of messages monitored by the at**
least one hook function;
- storing information related to an association between the
executable feature and corresponding graphics element and the
state of operation of the software application in a map data
structure to contain information related to at least one graphics
element for testing, the association and information being stored in
the map data structure during execution of the software application;

- automatically selecting from the map data structure at least one executable feature associated with a graphics element that has not been previously executed; and
- automatically executing the selected at least one executable feature.

Applicant respectfully disagrees with and traverses the rejection of claim 22, as amended.

Specifically, neither Parker nor Santee – whether consider alone or in any properly motivated combination - teaches or suggests installing at least one hook function into an application programming interface (API) of an operating system, the at least one hook function configured to monitor operating system messages communicated with the software application during execution of the software application being tested, as positively recited by claim 22, as amended. Also, neither Parker nor Santee – alone or in any properly motivated combination - teaches or suggests that at least some of the retrieved information descriptive of a state of operation of the software application retrieved by way of messages monitored by the at least one hook function, as positively recited by claim 22, as amended.

For at least the foregoing reasons, as well as for reasons analogous to those argued above in regard to claims 1 and 13 (as respectively amended), Applicant asserts that the § 103 rejection of claim 22, as amended, is invalid and should be withdrawn. Applicant further asserts that claim 22, as amended, is allowable.

Claims 23-28, as respectively amended, are also allowable as depending from an allowable base claim. While the respective § 103 rejections against claims 23-28 have been fully considered, they are not seen as contributing anything of merit.

1 **Conclusion**

2 The pending claims are in condition for allowance and action to that end is
3 respectfully requested. Should any issue remain that prevents allowance of the
4 application, the Office is encouraged to contact the undersigned prior to issuance
5 of a subsequent Office action.
6

7 Respectfully submitted,

8
9 Date: MARCH - 16 - 2007

10 By: Scott K. Gallert
11 Scott K. Gallert
12 Reg. No. 51,715
13 Lee & Hayes, PLLC
14 (509) 324-9256
15
16
17
18
19
20
21
22
23
24
25